CPSC 440/540: Advanced Machine Learning

Learning with Multivariate Gaussians

Danica Sutherland

University of British Columbia

Winter 2023

Couple of things

admin

- New slides format: let me know if something's worse about it
  - Or if things are going too fast – these slides are now closer to "old 540"
- Homework pushed back a day or two (deadline will be too)
- Project details also coming v. soon
- Final exam date has been set: **Saturday April 22 at noon**

Last Time: Multivariate Gaussians

- $X \sim \mathcal{N}(\mu, \Sigma)$ has $p(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} \det(\Sigma)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^{\mathsf{T}}\Sigma^{-1}(x-\mu)\right)$
  where $\mu \in \mathbb{R}^d$, $\Sigma \in \mathbb{R}^{d \times d}$ is symmetric with $\Sigma \succ 0$ (Σ is strictly positive definite)
  - If Σ is singular (so $\det(\Sigma) = 0$), degenerate Gaussian: supported on subspace of $\mathbb{R}^d$
- $\mathbb{E}[X] = \mu$ and $\mathrm{Cov}(X) = \Sigma$, i.e. $\mathrm{Cov}(X_j, X_{j'}) = \Sigma_{jj'}$.

Last Time: Multivariate Gaussians

- $X \sim \mathcal{N}(\mu, \Sigma)$ has $p(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} \det(\Sigma)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^{\mathsf{T}}\Sigma^{-1}(x-\mu)\right)$
  where $\mu \in \mathbb{R}^d$, $\Sigma \in \mathbb{R}^{d\times d}$ is symmetric with $\Sigma \succ 0$ (Σ is strictly positive definite)
  - If Σ is singular (so $\det(\Sigma) = 0$), degenerate Gaussian: supported on subspace of $\mathbb{R}^d$
- $\mathbb{E}[X] = \mu$ and $\mathrm{Cov}(X) = \Sigma$, i.e. $\mathrm{Cov}(X_j, X_{j'}) = \Sigma_{jj'}$.
- $AX + b \sim \mathcal{N}(A\mu + b, A\Sigma A^{\mathsf{T}})$

$X \sim \mathcal{N}(\mu, \Sigma)$

$$v^{T}X \sim \mathcal{N}\Big(v^{T}\mu, \underbrace{v^{T}\Sigma v}_{\geq 0 \text{ iff } \Sigma \succeq 0}\Big)$$

psd

Last Time: Multivariate Gaussians

- $X \sim \mathcal{N}(\mu, \Sigma)$ has $p(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} \det(\Sigma)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^\mathsf{T}\Sigma^{-1}(x-\mu)\right)$ where $\mu \in \mathbb{R}^d$, $\Sigma \in \mathbb{R}^{d\times d}$ is symmetric with $\Sigma \succ 0$ (Σ is strictly positive definite)
  - If Σ is singular (so $\det(\Sigma) = 0$), degenerate Gaussian: supported on subspace of $\mathbb{R}^d$
- $\mathbb{E}[X] = \mu$ and $\mathrm{Cov}(X) = \Sigma$, i.e. $\mathrm{Cov}(X_j, X_{j'}) = \Sigma_{jj'}$.
- $AX + b \sim \mathcal{N}(A\mu + b, A\Sigma A^\mathsf{T})$
- Marginalizing: if $\begin{bmatrix} X \\ Z \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_X \\ \mu_Z \end{bmatrix}, \begin{bmatrix} \Sigma_{XX} & \Sigma_{XZ} \\ \Sigma_{ZX} & \Sigma_{ZZ} \end{bmatrix}\right)$, then $X \sim \mathcal{N}(\mu_X, \Sigma_{XX})$
- Conditioning: $X \mid Z \sim \mathcal{N}(\mu_X + \Sigma_{XZ}\Sigma_{ZZ}^{-1}(Z - \mu_Z),\ \Sigma_{XX} - \Sigma_{XZ}\Sigma_{ZZ}^{-1}\Sigma_{ZX})$
- Implies $X_j \perp\!\!\!\perp X_{j'}$ iff $\Sigma_{jj'} = 0$

Conditional Independence in Gaussians

- Independence in Gaussians is determined by sparsity pattern of the covariance Σ.
  - Sparsity pattern: "where the non-zeroes are".
  - $X_i \perp\!\!\!\perp X_j$ iff $\Sigma_{ij} = 0$.

Conditional Independence in Gaussians

- Independence in Gaussians is determined by sparsity pattern of the covariance Σ.
  - Sparsity pattern: "where the non-zeroes are".
  - $X_i \perp\!\!\!\perp X_j$ iff $\Sigma_{ij} = 0$.

- Gaussians' conditional independence: sparsity of the precision matrix, $\Theta \triangleq \Sigma^{-1}$.
  - $X_i \perp\!\!\!\perp X_j \mid \{X_k : k \notin \{i, j\}\}$ iff $\Theta_{ij} = 0$.

Conditional Independence in Gaussians

- Independence in Gaussians is determined by sparsity pattern of the covariance Σ.
  - Sparsity pattern: "where the non-zeroes are".
  - $X_i \perp\!\!\!\perp X_j$ iff $\Sigma_{ij} = 0$.

- Gaussians' conditional independence: sparsity of the precision matrix, $\Theta \triangleq \Sigma^{-1}$.
  - $X_i \perp\!\!\!\perp X_j \mid \{X_k : k \notin \{i, j\}\}$ iff $\Theta_{ij} = 0$.

- We use the sparsity pattern of Θ to define a graph.
  - Each node in the graph corresponds to a variable $j \in \{1, 2, \ldots, d\}$.
  - Each edge in the graph corresponds to a non-zero Θ_{ij}.

Conditional Independence in Gaussians

- Independence in Gaussians is determined by sparsity pattern of the covariance Σ.
  - Sparsity pattern: "where the non-zeroes are".
  - $X_i \perp\!\!\!\perp X_j$ iff $\Sigma_{ij} = 0$.

- Gaussians' conditional independence: sparsity of the precision matrix, $\Theta \triangleq \Sigma^{-1}$.
  - $X_i \perp\!\!\!\perp X_j \mid \{X_k : k \notin \{i, j\}\}$ iff $\Theta_{ij} = 0$.

- We use the sparsity pattern of Θ to define a graph.
  - Each node in the graph corresponds to a variable $j \in \{1, 2, \ldots, d\}$.
  - Each edge in the graph corresponds to a non-zero Θ_{ij}.

- Checking independence and conditional independence using the graph:
  - $X_i \perp\!\!\!\perp X_j$ if no path exists between X_i and X_j in the graph.
  - $X_i \perp\!\!\!\perp X_j \mid X_k$ if X_k blocks all paths from X_i to X_j in the graph.
    - Technically, this only checks whether independence is implied by the sparsity pattern.

Conditional Independence in Gaussians

- Consider a Gaussian with the following covariance matrix:

$$\Sigma = \begin{bmatrix} 0.0494 & -0.0444 & -0.0312 & 0.0034 & -0.0010 \\ -0.0444 & 0.1083 & 0.0761 & -0.0083 & 0.0025 \\ -0.0312 & 0.0761 & 0.1872 & -0.0204 & 0.0062 \\ 0.0034 & -0.0083 & -0.0204 & 0.0528 & -0.0159 \\ -0.0010 & 0.0025 & 0.0062 & -0.0159 & 0.2636 \end{bmatrix}$$

- $\Sigma_{ij} \neq 0$, so all variables are dependent: $X_1 \not\perp X_2$, $X_1 \not\perp X_5$, and so on.
  - This would show up in graph: you'd be able to reach any X_i from any X_j.

Conditional Independence in Gaussians

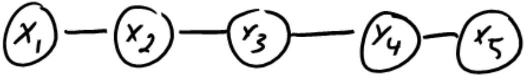
- Consider a Gaussian with the following covariance matrix:

$$\Sigma = \begin{bmatrix} 0.0494 & -0.0444 & -0.0312 & 0.0034 & -0.0010 \\ -0.0444 & 0.1083 & 0.0761 & -0.0083 & 0.0025 \\ -0.0312 & 0.0761 & 0.1872 & -0.0204 & 0.0062 \\ 0.0034 & -0.0083 & -0.0204 & 0.0528 & -0.0159 \\ -0.0010 & 0.0025 & 0.0062 & -0.0159 & 0.2636 \end{bmatrix}$$

- $\Sigma_{ij} \neq 0$, so all variables are dependent: $X_1 \not\perp X_2$, $X_1 \not\perp X_5$, and so on.
    - This would show up in graph: you'd be able to reach any X_i from any X_j.
- The inverse of this particular Σ is a tri-diagonal matrix:

$$\Sigma^{-1} = \begin{bmatrix} 32.0897 & 13.1740 & 0 & 0 & 0 \\ 13.1740 & 18.3444 & -5.2602 & 0 & 0 \\ 0 & -5.2602 & 7.7173 & 2.1597 & 0 \\ 0 & 0 & 2.1597 & 20.1232 & 1.1670 \\ 0 & 0 & 0 & 1.1670 & 3.8644 \end{bmatrix}$$

- So conditional independence is described by a 5-node "chain'-structured" graph:



Conditional Independence in Gaussians

- All variables are dependent in this graph, since a path exists.

Conditional Independence in Gaussians

- All variables are dependent in this graph, since a path exists.

- But we have many conditional independences such as:
  - $X_1 \perp\!\!\!\perp X_3 \mid X_2$.

Conditional Independence in Gaussians

- All variables are dependent in this graph, since a path exists.

- But we have many conditional independences such as:
  - $X_1 \perp\!\!\!\perp X_3 \mid X_2$.
  - $X_2 \perp\!\!\!\perp X_5 \mid X_4$.

Conditional Independence in Gaussians

- All variables are dependent in this graph, since a path exists.

- But we have many conditional independences such as:
  - $X_1 \perp\!\!\!\perp X_3 \mid X_2$.
  - $X_2 \perp\!\!\!\perp X_5 \mid X_4$.
  - $X_1 \perp\!\!\!\perp X_5 \mid X_3$.

Conditional Independence in Gaussians

- All variables are dependent in this graph, since a path exists.

- But we have many conditional independences such as:
  - $X_1 \perp\!\!\!\perp X_3 \mid X_2$.
  - $X_2 \perp\!\!\!\perp X_5 \mid X_4$.
  - $X_1 \perp\!\!\!\perp X_5 \mid X_3$.
  - $X_1 \perp\!\!\!\perp X_3, X_4, X_5 \mid X_2$ (the "Markov property").

Conditional Independence in Gaussians

- All variables are dependent in this graph, since a path exists.

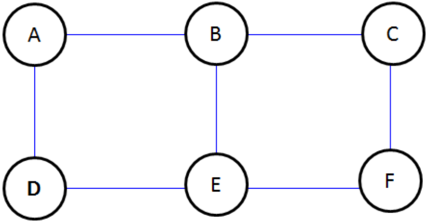
- But we have many conditional independences such as:
  - $X_1 \perp\!\!\!\perp X_3 \mid X_2$.
  - $X_2 \perp\!\!\!\perp X_5 \mid X_4$.
  - $X_1 \perp\!\!\!\perp X_5 \mid X_3$.
  - $X_1 \perp\!\!\!\perp X_3, X_4, X_5 \mid X_2$ (the "Markov property").
  - $X_1, X_2 \perp\!\!\!\perp X_4, X_5 \mid X_3$.

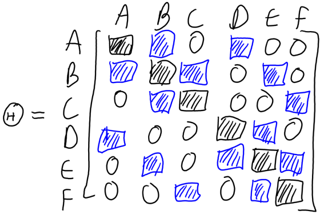
Conditional Independence in Gaussians

- Checking conditional independence among variable groups in Gaussians:
  - $A \perp\!\!\!\perp B \mid C$ if C blocks all paths from any A to any B.

Conditional Independence in Gaussians

- Checking conditional independence among variable groups in Gaussians:
  - $A \perp\!\!\!\perp B \mid C$ if C blocks all paths from any A to any B.

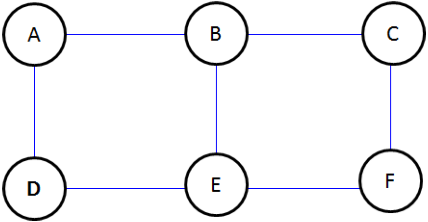


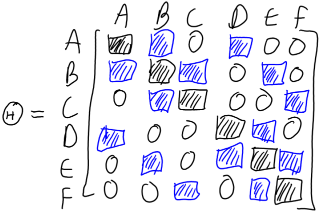


- Example:
  - $A \not\perp\!\!\!\perp C$.

Conditional Independence in Gaussians

- Checking conditional independence among variable groups in Gaussians:
  - $A \perp\!\!\!\perp B \mid C$ if C blocks all paths from any A to any B.

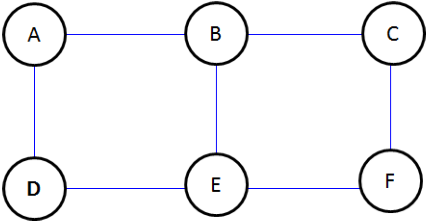


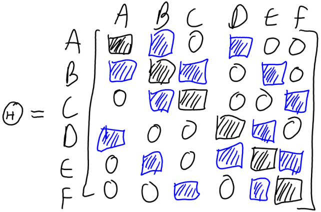


- Example:
  - $A \not\perp\!\!\!\perp C$.
  - $A \not\perp\!\!\!\perp C \mid B$.

Conditional Independence in Gaussians

- Checking conditional independence among variable groups in Gaussians:
  - $A \perp\!\!\!\perp B \mid C$ if C blocks all paths from any A to any B.

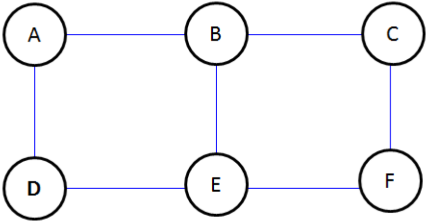


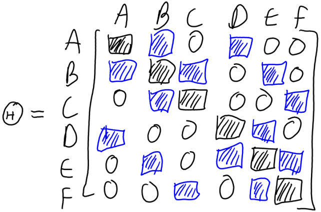


- Example:
  - $A \not\perp\!\!\!\perp C$.
  - $A \not\perp\!\!\!\perp C \mid B$.
  - $A \perp\!\!\!\perp C \mid B, E$.

Conditional Independence in Gaussians

- Checking conditional independence among variable groups in Gaussians:
  - $A \perp\!\!\!\perp B \mid C$ if C blocks all paths from any A to any B.

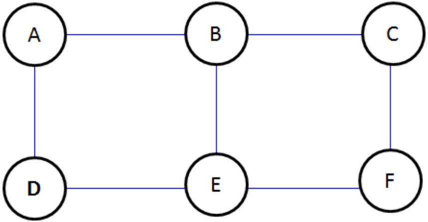


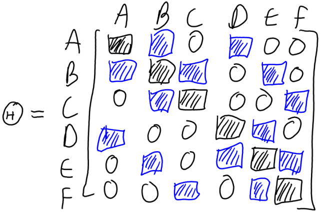


- Example:
  - $A \not\perp\!\!\!\perp C$.
  - $A \not\perp\!\!\!\perp C \mid B$.
  - $A \perp\!\!\!\perp C \mid B, E$.
  - $A, B \not\perp\!\!\!\perp F \mid C$

Conditional Independence in Gaussians

- Checking conditional independence among variable groups in Gaussians:
  - $A \perp\!\!\!\perp B \mid C$ if C blocks all paths from any A to any B.





- Example:
  - $A \not\perp\!\!\!\perp C$.
  - $A \not\perp\!\!\!\perp C \mid B$.
  - $A \perp\!\!\!\perp C \mid B, E$.
  - $A, B \not\perp\!\!\!\perp F \mid C$
  - $A, B \perp\!\!\!\perp F \mid C, E$.

Discussion of Independence in Gaussians

- If Σ is diagonal then Θ is diagonal.
  - This gives a disconnected graph: all variables are independent.

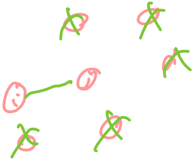
Discussion of Independence in Gaussians

- If Σ is diagonal then Θ is diagonal.
  - This gives a disconnected graph: all variables are independent.
- If Θ is a full matrix, graph does not imply any conditional independences.
  - "Everything depends on everything, no matter how many of the X_j you know."

Discussion of Independence in Gaussians

- If Σ is diagonal then Θ is diagonal.
  - This gives a disconnected graph: all variables are independent.
- If Θ is a full matrix, graph does not imply any conditional independences.
  - "Everything depends on everything, no matter how many of the X_j you know."
- Dependencies can exist if $\Theta_{ij} = 0$ due to correlations with other variables.
  - Only independent if all paths that correlation could go across are blocked.

$\Theta_{ij} = 0$ iff $X_i \perp\!\!\!\perp X_j \mid \{X_k : k \notin \{i,j\}\}$



Conditional Independence and the Precision Matrix

cond. ind.

- Where does the connection of $\cancel{\mathcal{G}}$ to the precision matrix come from?

Conditional Independence and the Precision Matrix

- Where does the connection of ~~0~~ to the precision matrix come from?
- Let's use linear regression to predict X_i from $\{X_k : k \neq i, k \neq j\}$
- Define $R_{i,\neg j}$ as the residual, $X_i - \sum_{k \notin \{i,j\}} w_k X_k - b$

Conditional Independence and the Precision Matrix

- Where does the connection of Θ to the precision matrix come from?
- Let's use linear regression to predict X_i from $\{X_k : k \neq i, k \neq j\}$
- Define $R_{i,\neg j}$ as the residual, $X_i - \sum_{k \notin \{i,j\}} w_k X_k - b$
- The partial correlation coefficient is the correlation between $R_{i,\neg j}$ and $R_{j,\neg i}$

Conditional Independence and the Precision Matrix

- Where does the connection of Θ to the precision matrix come from?
- Let's use linear regression to predict X_i from $\{X_k : k \neq i, k \neq j\}$
- Define $R_{i,\neg j}$ as the residual, $X_i - \sum_{k \notin \{i,j\}} w_k X_k - b$
- The partial correlation coefficient is the correlation between $R_{i,\neg j}$ and $R_{j,\neg i}$
  - Can work out that it's exactly $-\Theta_{ij}/\sqrt{\Theta_{ii}\Theta_{jj}}$
  - Thus partial correlation coefficient is 0 iff $\Theta_{ij} = 0$
- In Gaussians, dependencies are linear: zero partial correlation iff conditionally independent

Outline

1. Conditional Independence
2. Learning in Multivariate Gaussians
3. Supervised Learning with Gaussians
4. Bayesian Linear Regression
5. Rejection and Importance Sampling

MLE for Multivariate Gaussian (Mean Vector)

- If $x^i \overset{iid}{\sim} \mathcal{N}(\mu, \Sigma)$, we have

$$p(x^i \mid \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x^i - \mu)^\top \Sigma^{-1}(x^i - \mu)\right),$$

so up to a constant our negative log-likelihood for n examples is

$$\frac{1}{2}\sum_{i=1}^{n}(x^i - \mu)^\top \Sigma^{-1}(x^i - \mu) + \frac{n}{2}\log|\Sigma|.$$

MLE for Multivariate Gaussian (Mean Vector)

- If $x^i \overset{iid}{\sim} \mathcal{N}(\mu, \Sigma)$, we have

$$p(x^i \mid \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x^i - \mu)^\top \Sigma^{-1}(x^i - \mu)\right),$$

$\nabla_x x^T A x = 2Ax$

so up to a constant our negative log-likelihood for n examples is

$$\frac{1}{2}\sum_{i=1}^{n}(x^i - \mu)^\top \Sigma^{-1}(x^i - \mu) + \frac{n}{2}\log|\Sigma|.$$

$\nabla_\mu = \sum_i \Sigma^{-1}(x^i - \mu) = 0$

$\Sigma^{-1}\left(\frac{1}{n}\sum_i x^i - \mu\right)$

- This is a convex quadratic in μ. Setting gradient to zero gives

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x^i.$$

  - MLE for μ is the mean along each dimension, and it does not depend on Σ.

MLE for Multivariate Gaussians (Covariance Matrix)

- To get MLE for Σ we can re-parameterize in terms of precision matrix $\Theta = \Sigma^{-1}$,

$$\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Sigma^{-1}(x^i-\mu) + \frac{n}{2}\log\det\Sigma$$

$$\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$$

$$= \frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Theta(x^i-\mu) + \frac{n}{2}\log\det\Theta^{-1}$$

$$\underbrace{(x^i-\mu)^\top\Theta}_{1\times d}\qquad 1\times d \quad d\times d \quad d\times 1$$

$$= n\left[\frac{1}{2n}\sum_i \mathrm{Tr}\left((x^i-\mu)^T\theta(x^i-\mu)\right) + \frac{1}{2}\log\det\theta^{-1}\right]$$

$$= \frac{1}{2}\mathrm{Tr}\left(\underbrace{\frac{1}{n}\sum_i (x^i-\mu)(x^i-\mu)^T}\theta\right) - \frac{1}{2}\log\det\theta\Big]$$

$$= \frac{n}{2}\left[\mathrm{Tr}(S\theta) - \log\det\theta\right]$$

$$\mathrm{Tr}(AB) = \sum_i (AB)_{ii} = \sum_i\sum_j A_{ij}B_{ji} = (A * B^T).\mathrm{sum}()$$

MLE for Multivariate Gaussians (Covariance Matrix)

- To get MLE for Σ we can re-parameterize in terms of precision matrix $\Theta = \Sigma^{-1}$,

$$
\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Sigma^{-1}(x^i-\mu)+\frac{n}{2}\log\det\Sigma \\
=&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Theta(x^i-\mu)+\frac{n}{2}\log\det\Theta^{-1}
\end{aligned}
$$

- After some work (bonus slides), we obtain that this is equal to

$$
f(\Theta)=\frac{n}{2}\mathrm{Tr}(\mathbf{S}\Theta)-\frac{n}{2}\log\det\Theta, \text{ with } \mathbf{S}=\frac{1}{n}\sum_{i=1}^{n}(x^i-\mu)(x^i-\mu)^\top
$$

where:

- $\mathbf{S}$ is the sample covariance: if $\tilde{\mathbf{X}} = \mathbf{X} - \mu\mathbf{1}\mu^\mathsf{T}$ is centred data, $S = (1/n)\tilde{\mathbf{X}}^\mathsf{T}\tilde{\mathbf{X}}$.
- Trace operator $\mathrm{Tr}(\mathbf{A})$ is the sum of the diagonal elements of $\mathbf{A}$.

MLE for Multivariate Gaussians (Covariance Matrix)

- Gradient matrix of NLL with respect to Θ is (not obvious)

$$\nabla f(\Theta) = \frac{n}{2}\mathbf{S} - \frac{n}{2}\Theta^{-1}.$$

$\nabla_\Theta \log\det\Theta = \Theta^{-1}$

$\frac{d}{dx}\log|x| = \frac{1}{|x|}$

MLE for Multivariate Gaussians (Covariance Matrix)

- Gradient matrix of NLL with respect to Θ is (not obvious)

$$\nabla f(\Theta) = \frac{n}{2}\mathbf{S} - \frac{n}{2}\Theta^{-1}.$$

- The MLE for a given μ is obtained by setting gradient matrix to zero, giving

$$\Theta = \mathbf{S}^{-1} \quad \text{or} \quad \Sigma = \frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^\top.$$

MLE for Multivariate Gaussians (Covariance Matrix)

- Gradient matrix of NLL with respect to Θ is (not obvious)

$$\nabla f(\Theta) = \frac{n}{2}\mathbf{S} - \frac{n}{2}\Theta^{-1}.$$

- The MLE for a given μ is obtained by setting gradient matrix to zero, giving

$$\Theta = \mathbf{S}^{-1} \quad \text{or} \quad \Sigma = \frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^\top.$$

- The constraint $\Sigma \succ 0$ means we need positive-definite sample covariance, $S \succ 0$.
  - If S is not positive-definite, NLL is unbounded below and MLE doesn't exist.
  - This is like requiring "not all values are the same" in univariate Gaussian.
    - In d-dimensions, you need d linearly independent x^i values (no "multi-collinearity")

if
$\tilde{X} = X - 1\mu^T$

$S = \frac{1}{n}\tilde{X}^T\tilde{X}$

$d \times n$ $n \times d$

MLE for Multivariate Gaussians (Covariance Matrix)

- Gradient matrix of NLL with respect to Θ is (not obvious)

$$\nabla f(\Theta) = \frac{n}{2}\mathbf{S} - \frac{n}{2}\Theta^{-1}.$$

- The MLE for a given μ is obtained by setting gradient matrix to zero, giving

$$\Theta = \mathbf{S}^{-1} \quad \text{or} \quad \Sigma = \frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^\top.$$

- The constraint $\Sigma \succ 0$ means we need positive-definite sample covariance, $S \succ 0$.
  - If S is not positive-definite, NLL is unbounded below and MLE doesn't exist.
  - This is like requiring "not all values are the same" in univariate Gaussian.
    - In d-dimensions, you need d linearly independent x^i values (no "multi-collinearity")

- Note: most distributions' MLEs don't do "moment matching" like this.

MAP Estimation for Mean

- For fixed Σ, conjugate prior for mean is a Gaussian:

$$x^i \sim \mathcal{N}(\mu, \Sigma) \qquad \mu \sim \mathcal{N}(\mu_0, \Sigma_0) \quad \text{implies} \quad \mu \mid X, \Sigma \sim \mathcal{N}(\mu^+, \Sigma^+),$$

MAP Estimation for Mean

- For fixed Σ, conjugate prior for mean is a Gaussian:

$$x^i \sim \mathcal{N}(\mu, \Sigma) \qquad \mu \sim \mathcal{N}(\mu_0, \Sigma_0) \quad \text{implies} \quad \mu \mid X, \Sigma \sim \mathcal{N}(\mu^+, \Sigma^+),$$

where (using product of Gaussians property we are about to cover)

$$\begin{aligned}
\Sigma^+ &= (n\Sigma^{-1} + \Sigma_0^{-1})^{-1}, \\
\mu^+ &= \Sigma^+(n\Sigma^{-1}\mu_{\textsf{MLE}} + \Sigma_0^{-1}\mu_0).
\end{aligned}$$

MAP estimate of μ

MAP Estimation for Mean

- For fixed Σ, conjugate prior for mean is a Gaussian:

$$x^i \sim \mathcal{N}(\mu, \Sigma) \qquad \mu \sim \mathcal{N}(\mu_0, \Sigma_0) \quad \text{implies} \quad \mu \mid X, \Sigma \sim \mathcal{N}(\mu^+, \Sigma^+),$$

where (using product of Gaussians property we are about to cover)

$$\begin{aligned}\Sigma^+ &= (n\Sigma^{-1} + \Sigma_0^{-1})^{-1},\\ \mu^+ &= \Sigma^+(n\Sigma^{-1}\mu_{\text{MLE}} + \Sigma_0^{-1}\mu_0).\end{aligned}$$

MAP estimate of μ

- In special case of $\Sigma = \sigma^2\mathbf{I}$ and $\Sigma_0 = (1/\lambda)\mathbf{I}$, we get

$$\begin{aligned}\Sigma^+ &= ((n/\sigma^2)\mathbf{I} + \lambda\mathbf{I})^{-1} = \frac{1}{\frac{1}{\sigma^2/n} + \frac{1}{\lambda}}\mathbf{I},\\ \mu^+ &= \Sigma^+(\frac{n}{\sigma^2}\mu_{\text{MLE}} + \lambda\mu_0).\end{aligned}$$

MAP Estimation for Mean

- For fixed Σ, conjugate prior for mean is a Gaussian:

$$x^i \sim \mathcal{N}(\mu, \Sigma) \qquad \mu \sim \mathcal{N}(\mu_0, \Sigma_0) \quad \text{implies} \quad \mu \mid X, \Sigma \sim \mathcal{N}(\mu^+, \Sigma^+),$$

where (using product of Gaussians property we are about to cover)

$$\Sigma^+ = (n\Sigma^{-1} + \Sigma_0^{-1})^{-1},$$

$$\mu^+ = \Sigma^+(n\Sigma^{-1}\mu_{\text{MLE}} + \Sigma_0^{-1}\mu_0).$$

MAP estimate of μ

- In special case of $\Sigma = \sigma^2\mathbf{I}$ and $\Sigma_0 = (1/\lambda)\mathbf{I}$, we get

$$\Sigma^+ = ((n/\sigma^2)\mathbf{I} + \lambda\mathbf{I})^{-1} = \frac{1}{\frac{1}{\sigma^2/n} + \frac{1}{\lambda}}\mathbf{I},$$

$$\mu^+ = \Sigma^+(\frac{n}{\sigma^2}\mu_{\text{MLE}} + \lambda\mu_0).$$

- Posterior predictive is $\mathcal{N}(\mu^+, \Sigma + \Sigma^+)$ – take product of $(n+2)$ then marginalize.
  - Many Bayesian inference tasks have closed form, or Monte Carlo is easy.

Product of Gaussian Densities Property

- Consider variable x whose PDF is written as product of two Gaussians,

$$p(x) = f_1(x)f_2(x)$$

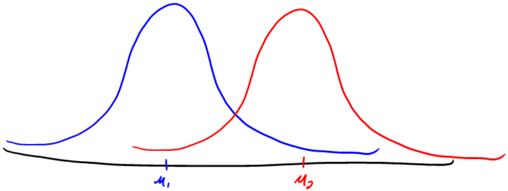
  where:
  - f_1 is proportional to a Gaussian density with mean μ_1 and covariance $\mathbf{I}$.
  - f_2 is proportional to a Gaussian density with mean μ_2 and covariance $\mathbf{I}$.

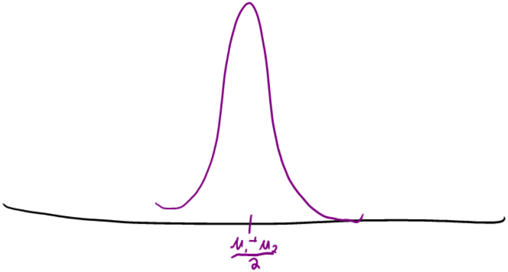
Product of Gaussian Densities Property

- Consider variable x whose PDF is written as product of two Gaussians,

$$p(x) = f_1(x) f_2(x)$$

  where:
  - f_1 is proportional to a Gaussian density with mean μ_1 and covariance $\mathbf{I}$.
  - f_2 is proportional to a Gaussian density with mean μ_2 and covariance $\mathbf{I}$.
- Then this product of Gaussian PDFs is a Gaussian with $\mu = \frac{\mu_1 + \mu_2}{2}$ and $\Sigma = \frac{1}{2}.\mathbf{I}$





Product of Gaussian Densities Property

- If $p(x) \propto f_1(x) f_2(x)$ with
  - f_1 proportional to a Gaussian with mean μ_1 and covariance Σ_1.
  - f_2 proportional to a Gaussian with mean μ_2 and covariance Σ_2.

Product of Gaussian Densities Property

- If $p(x) \propto f_1(x) f_2(x)$ with
  - f_1 proportional to a Gaussian with mean μ_1 and covariance Σ_1.
  - f_2 proportional to a Gaussian with mean μ_2 and covariance Σ_2.
- Then p is a Gaussian with (see PML2 2.2.7.6)

$$\text{covariance } \Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}.$$

Product of Gaussian Densities Property

- If $p(x) \propto f_1(x) f_2(x)$ with
  - f_1 proportional to a Gaussian with mean μ_1 and covariance Σ_1.
  - f_2 proportional to a Gaussian with mean μ_2 and covariance Σ_2.
- Then p is a Gaussian with (see PML2 2.2.7.6)

$$\text{covariance} \;\; \Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}.$$

$$\text{mean} \;\; \mu = \Sigma\Sigma_1^{-1}\mu_1 + \Sigma\Sigma_2^{-1}\mu_2,$$

Product of Gaussian Densities Property

- If $p(x) \propto f_1(x) f_2(x)$ with
  - f_1 proportional to a Gaussian with mean μ_1 and covariance Σ_1.
  - f_2 proportional to a Gaussian with mean μ_2 and covariance Σ_2.
- Then p is a Gaussian with (see PML2 2.2.7.6)

$$\text{covariance} \quad \Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}.$$

$$\text{mean} \quad \mu = \Sigma\Sigma_1^{-1}\mu_1 + \Sigma\Sigma_2^{-1}\mu_2,$$

- How we do we use this to derive the posterior distribution for the mean?

$$\begin{aligned} p(\mu \mid \mathbf{X}, \Sigma, \mu_0, \Sigma_0) &\propto p(\mu \mid \mu_0, \Sigma_0) \prod_{i=1}^{n} p(x^i \mid \mu, \Sigma) && \text{(Bayes rule)} \\ &= p(\mu \mid \mu_0, \Sigma_0) \prod_{i=1}^{n} p(\mu \mid x^i, \Sigma) && \text{(symmetry of } x^i \text{ and } \mu) \\ &= \text{(product of } (n+1) \text{ Gaussians).} \end{aligned}$$

MAP Estimation in Multivariate Gaussian (Trace Regularization)

- A common MAP estimate for Σ is

$$\hat{\Sigma} = \mathbf{S} + \lambda \mathbf{I},$$

  where S is the covariance of the data.
  - Key advantage: $\hat{\Sigma}$ is positive-definite (eigenvalues are at least λ).

MAP Estimation in Multivariate Gaussian (Trace Regularization)

- A common MAP estimate for Σ is

$$\hat{\Sigma} = \mathbf{S} + \lambda \mathbf{I},$$

  where S is the covariance of the data.
  - Key advantage: $\hat{\Sigma}$ is positive-definite (eigenvalues are at least λ).

- This corresponds to L1 regularization of precision diagonals (see bonus)

$$f(\Theta) = \underbrace{\text{Tr}(\mathbf{S}\Theta) - \log\det\Theta}_{\text{NLL times } 2/n} + \lambda \sum_{j=1}^{d} |\Theta_{jj}|.$$

  Note it doesn't set Θ_{jj} values to exactly zero.
  - Log-determinant term becomes arbitrarily steep as the Θ_{jj} approach 0.

Graphical LASSO

- A popular generalization called the graphical LASSO,

$$f(\Theta) = \mathrm{Tr}(\mathbf{S}\Theta) - \log\det\Theta + \lambda \sum_{i=1}^{d} \sum_{j=1}^{d} |\Theta_{ij}|,$$

where we apply L1 regularization to all elements of Θ.

Graphical LASSO

- A popular generalization called the graphical LASSO,

$$f(\Theta) = \text{Tr}(\mathbf{S}\Theta) - \log\det\Theta + \lambda\sum_{i=1}^{d}\sum_{j=1}^{d}|\Theta_{ij}|,$$

  where we apply L1 regularization to all elements of Θ.

- With large enough λ, gives sparse off-diagonals in Θ.
  - Need specialized optimization algorithms to solve this problem.

Graphical LASSO

- A popular generalization called the graphical LASSO,

$$f(\Theta) = \text{Tr}(\mathbf{S}\Theta) - \log\det\Theta + \lambda\sum_{i=1}^{d}\sum_{j=1}^{d}|\Theta_{ij}|,$$

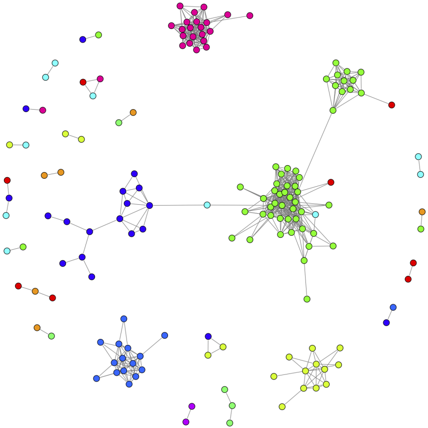
  where we apply L1 regularization to all elements of Θ.

- With large enough λ, gives sparse off-diagonals in Θ.
  - Need specialized optimization algorithms to solve this problem.

- Recall that sparsity of Θ determines conditional independence.
  - When we set a $\Theta_{ij} = 0$ it remove an edges from the graph.
    - Makes the graph simpler, and can make computations cheaper.

Graphical LASSO Example

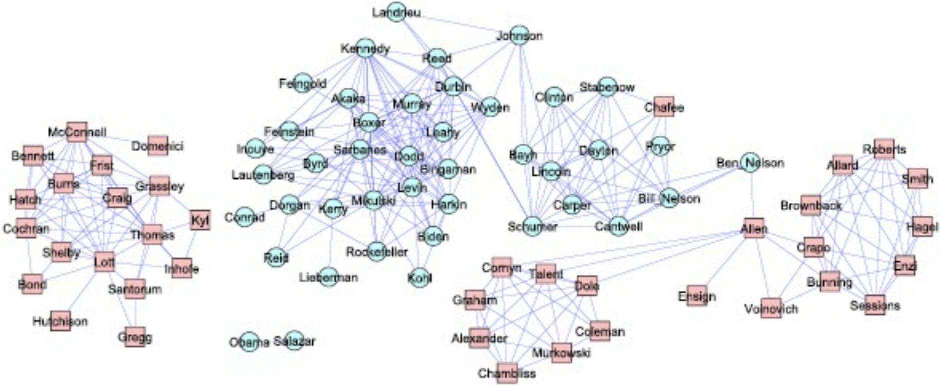
- Graphical LASSO applied to stocks data:



https://normaldeviate.wordpress.com/2012/09/17/high-dimensional-undirected-graphical-models

Graphical LASSO Example

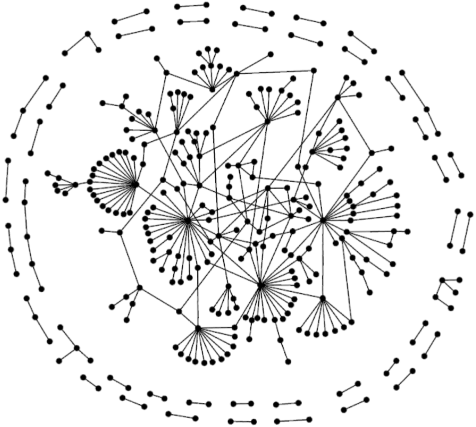
- Graphical LASSO applied to US senate voting data (Bush junior era):



https://normaldeviate.wordpress.com/2012/09/17/high-dimensional-undirected-graphical-models

Graphical LASSO Example

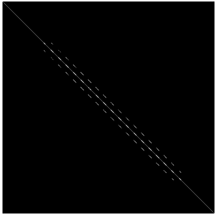
- Graphical LASSO applied to protein data:



https://normaldeviate.wordpress.com/2012/09/17/high-dimensional-undirected-graphical-models

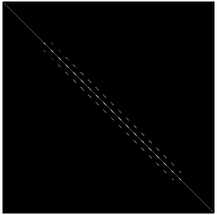
Graphical LASSO on Digits

- Precision matrix from graphical LASSO applied to MNIST digits ($\lambda = 1/8$):



Graphical LASSO on Digits

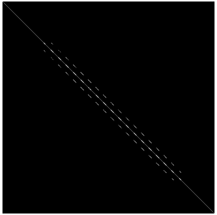
- Precision matrix from graphical LASSO applied to MNIST digits ($\lambda = 1/8$):



- To understand this picture, first the size of the precision matrix:
  - The images of digits, which are $m \times m$ matrices (m pixels by m pixels)
    - This gives $d = m^2$ elements of x^i, which we'll assume are in "column-major" order.
    - Frist m elements of x^i are column 1, next m elements are columm 2, and so on.

Graphical LASSO on Digits

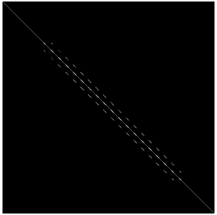
- Precision matrix from graphical LASSO applied to MNIST digits ($\lambda = 1/8$):



- To understand this picture, first the size of the precision matrix:
  - The images of digits, which are $m \times m$ matrices (m pixels by m pixels)
    - This gives $d = m^2$ elements of x^i, which we'll assume are in "column-major" order.
    - Frist m elements of x^i are column 1, next m elements are columm 2, and so on.
  - The picture above, which is $d \times d$ so will thus be $m^2 \times m^2$.

Graphical LASSO on Digits

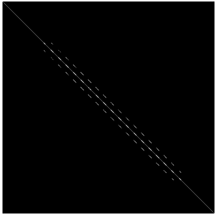
- Precision matrix from graphical LASSO applied to MNIST digits ($\lambda = 1/8$):



- So what are the non-zeroes in the precision matrix?
  1. The diagonals $\Theta_{i,i}$ (positive-definite matrices must have positive diagonals).

Graphical LASSO on Digits

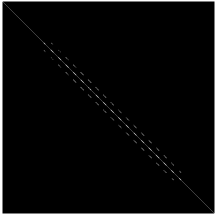
- Precision matrix from graphical LASSO applied to MNIST digits ($\lambda = 1/8$):



- So what are the non-zeroes in the precision matrix?
  1. The diagonals $\Theta_{i,i}$ (positive-definite matrices must have positive diagonals).
  2. The first off-diagonals $\Theta_{i,i+1}$ and $\Theta_{i+1,i}$.
     - This represents the dependencies between adjacent pixels vertically.

Graphical LASSO on Digits

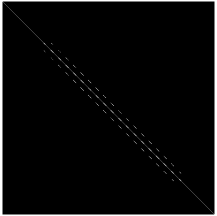
- Precision matrix from graphical LASSO applied to MNIST digits ($\lambda = 1/8$):



- So what are the non-zeroes in the precision matrix?
  1. The diagonals $\Theta_{i,i}$ (positive-definite matrices must have positive diagonals).
  2. The first off-diagonals $\Theta_{i,i+1}$ and $\Theta_{i+1,i}$.
     - This represents the dependencies between adjacent pixels vertically.
  3. The $(m+1)$ off-diagonals $\Theta_{i,i+m}$ and $\Theta_{i+m,i}$.
     - This represents the dependencies between adjacent pixels horizontally.
     - Because in "column-major" order, you go "right" a pixel every m indices.

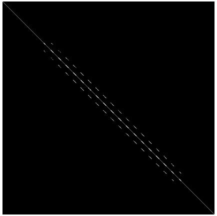
Graphical LASSO on Digits

- Precision matrix from graphical LASSO applied to MNIST digits ($\lambda = 1/8$):



Graphical LASSO on Digits

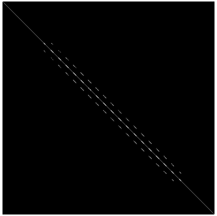
- Precision matrix from graphical LASSO applied to MNIST digits ($\lambda = 1/8$):



- The edges in the graph are pixels next to each other in the image.

Graphical LASSO on Digits

- Precision matrix from graphical LASSO applied to MNIST digits ($\lambda = 1/8$):



- The edges in the graph are pixels next to each other in the image.
- Graphical Lasso is a special case of structure learning in graphical models.
  - We will discusss graphical models more later.

Conjugate Priors for Covariance

bonus!

- Graphical LASSO is not using a conjugate prior.

Conjugate Priors for Covariance

bonus!

- Graphical LASSO is not using a conjugate prior.
- Conjugate prior for Θ with known mean is Wishart distribution
  - A multi-dimensional generalization of the gamma distribution.
    - Gamma is a distribution over positive scalars.
    - Wishart is a distribution over positive-definite matrices.

Conjugate Priors for Covariance

bonus!

- Graphical LASSO is not using a conjugate prior.
- Conjugate prior for Θ with known mean is Wishart distribution
  - A multi-dimensional generalization of the gamma distribution.
    - Gamma is a distribution over positive scalars.
    - Wishart is a distribution over positive-definite matrices.
  - Posterior predictive is a student t distribution.
  - Conjugate prior for Σ is inverse-Wishart (equivalent posterior).

Conjugate Priors for Covariance

bonus!

- Graphical LASSO is not using a conjugate prior.

- Conjugate prior for Θ with known mean is Wishart distribution
  - A multi-dimensional generalization of the gamma distribution.
    - Gamma is a distribution over positive scalars.
    - Wishart is a distribution over positive-definite matrices.
  - Posterior predictive is a student t distribution.
  - Conjugate prior for Σ is inverse-Wishart (equivalent posterior).

- If both μ and Θ are variables, conjugate prior is normal-Wishart.
  - Normal times Wishart, with a particular dependency among parameters.
  - Posterior predictive is again a student t distribution.

Conjugate Priors for Covariance

bonus!

- Graphical LASSO is not using a conjugate prior.
- Conjugate prior for Θ with known mean is Wishart distribution
  - A multi-dimensional generalization of the gamma distribution.
    - Gamma is a distribution over positive scalars.
    - Wishart is a distribution over positive-definite matrices.
  - Posterior predictive is a student t distribution.
  - Conjugate prior for Σ is inverse-Wishart (equivalent posterior).
- If both μ and Θ are variables, conjugate prior is normal-Wishart.
  - Normal times Wishart, with a particular dependency among parameters.
  - Posterior predictive is again a student t distribution.
- Wikipedia has already done a lot of possible homework questions for you:
  - `https://en.wikipedia.org/wiki/Conjugate_prior`

Outline

1. Conditional Independence
2. Learning in Multivariate Gaussians
3. Supervised Learning with Gaussians
4. Bayesian Linear Regression
5. Rejection and Importance Sampling

Generative Classification with Gaussians

- We previously considerd the generative classifier, naive Bayes.
  - Assumed $X_i \perp\!\!\!\perp X_j \mid Y$, which is strong/unrealistic.

Generative Classification with Gaussians

- We previously considerd the generative classifier, naive Bayes.
  - Assumed $X_i \perp\!\!\!\perp X_j \mid Y$, which is strong/unrealistic.

- Consider a generative classifier with continuous features:

$$
\begin{aligned}
p(y^i \mid x^i) &\propto p(x^i, y^i) \\
&= \underbrace{p(x^i \mid y^i)}_{\text{continuous}} \underbrace{p(y^i)}_{\text{discrete}} .
\end{aligned}
$$

Generative Classification with Gaussians

- We previously considerd the generative classifier, naive Bayes.
  - Assumed $X_i \perp\!\!\!\perp X_j \mid Y$, which is strong/unrealistic.

- Consider a generative classifier with continuous features:

$$\begin{aligned} p(y^i \mid x^i) &\propto p(x^i, y^i) \\ &= \underbrace{p(x^i \mid y^i)}_{\text{continuous}} \underbrace{p(y^i)}_{\text{discrete}}. \end{aligned}$$

- In Gaussian discriminant analysis (GDA) we assume $X \mid Y$ is Gaussian.
  - It's classification: output Y is categorical.
  - Classifier asks "which Gaussian makes this x^i most likely?"

Generative Classification with Gaussians

- We previously considerd the generative classifier, naive Bayes.
  - Assumed $X_i \perp\!\!\!\perp X_j \mid Y$, which is strong/unrealistic.

- Consider a generative classifier with continuous features:

$$\begin{aligned} p(y^i \mid x^i) &\propto p(x^i, y^i) \\ &= \underbrace{p(x^i \mid y^i)}_{\text{continuous}} \underbrace{p(y^i)}_{\text{discrete}} . \end{aligned}$$

- In Gaussian discriminant analysis (GDA) we assume $X \mid Y$ is Gaussian.
  - It's classification: output Y is categorical.
  - Classifier asks "which Gaussian makes this x^i most likely?"
  - This can model pairwise correlations within each class.
    - Doesn't need naive Bayes assumption.

Gaussian Discriminant Analysis (GDA) and Closed-Form MLE

- In Gaussian discriminant analysis we assume $X \mid Y$ is a Gaussian.

$$p(x^i, y^i = c) = \underbrace{p(y^i)\, p(x^i \mid y^i = c)}_{\text{product rule}} = \underbrace{\pi_c}_{\Pr(y^i=c)} \underbrace{p(x^i \mid \mu_c, \Sigma_c)}_{\text{Gaussian PDF}}.$$

Gaussian Discriminant Analysis (GDA) and Closed-Form MLE

- In Gaussian discriminant analysis we assume $X \mid Y$ is a Gaussian.

$$p(x^i, y^i = c) = \underbrace{p(y^i)\, p(x^i \mid y^i = c)}_{\text{product rule}} = \underbrace{\pi_c}_{\Pr(y^i = c)} \underbrace{p(x^i \mid \mu_c, \Sigma_c)}_{\text{Gaussian PDF}}.$$

- A special case is linear discriminant analysis (LDA):
  - Assume that Σ_c is the same for all classes c.

Gaussian Discriminant Analysis (GDA) and Closed-Form MLE

- In Gaussian discriminant analysis we assume $X \mid Y$ is a Gaussian.

$$p(x^i, y^i = c) = \underbrace{p(y^i)\, p(x^i \mid y^i = c)}_{\text{product rule}} = \underbrace{\pi_c}_{\Pr(y^i = c)} \underbrace{p(x^i \mid \mu_c, \Sigma_c)}_{\text{Gaussian PDF}}.$$

- A special case is linear discriminant analysis (LDA):
  - Assume that Σ_c is the same for all classes c.

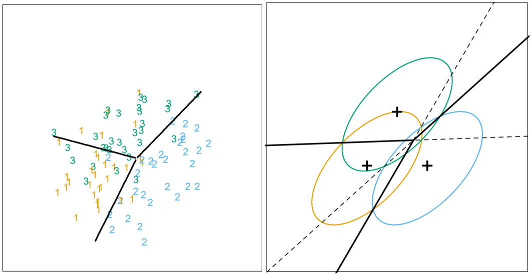
- In LDA the MLE has a simple closed-form expression:

$$\hat{\pi}_c = \frac{n_c}{n}, \quad \hat{\mu}_c = \frac{1}{n_c} \sum_{y^i = c} x^i.$$

  - $\hat{\pi}_c$ is fraction of times we are in class c; $\hat{\mu}$ is mean of class c.

Linear Discriminant Analysis (LDA)

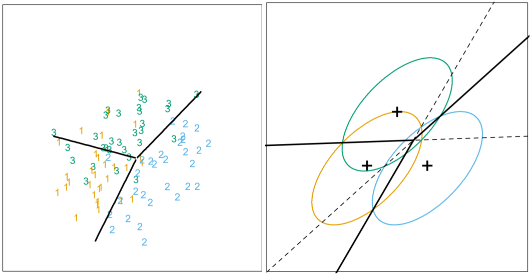
- Example of fitting linear discriminant analysis (LDA) to a 3-class problem:



https://web.stanford.edu/~hastie/Papers/ESLII.pdf

Linear Discriminant Analysis (LDA)

- Example of fitting linear discriminant analysis (LDA) to a 3-class problem:

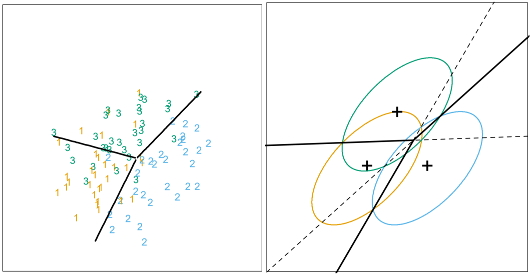


https://web.stanford.edu/~hastie/Papers/ESLII.pdf

- LDA is a linear classifier.
  - Unlike other linear classifiers (logistic regression, SVMs), it has a closed-form MLE.
  - Might not work well if assumptions (each class Gaussian, cov Σ) are bad fit to data.

Linear Discriminant Analysis (LDA)

- Example of fitting linear discriminant analysis (LDA) to a 3-class problem:



https://web.stanford.edu/~hastie/Papers/ESLII.pdf

- LDA is a linear classifier.
  - Unlike other linear classifiers (logistic regression, SVMs), it has a closed-form MLE.
  - Might not work well if assumptions (each class Gaussian, cov Σ) are bad fit to data.
- If class proportions π_c are equal, class label is determined by nearest mean.
  - Prediction is like in k-means clustering.

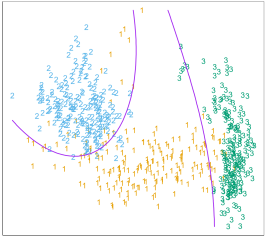
Gaussian Discriminant Analysis (GDA)

- We can also have a different covariance Σ_c for each class.
  - So the class will be determined by class proportions, means, and variances.

Gaussian Discriminant Analysis (GDA)

- We can also have a different covariance Σ_c for each class.
  - So the class will be determined by class proportions, means, and variances.
- The MLE for each each Σ_c is the covariance of data in class c,

$$\hat{\Sigma}_c = \frac{1}{n_c} \sum_{y^i = c} (x_i - \hat{\mu}_c)(x_i - \hat{\mu}_c)^T,$$



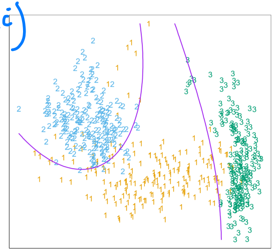
https://web.stanford.edu/~hastie/Papers/ESLII.pdf

Gaussian Discriminant Analysis (GDA)

- We can also have a different covariance Σ_c for each class.
  - So the class will be determined by class proportions, means, and variances.
- The MLE for each each Σ_c is the covariance of data in class c,

$$\hat{\Sigma}_c = \frac{1}{n_c}\sum_{y^i=c}(x_i - \hat{\mu}_c)(x_i - \hat{\mu}_c)^T,$$

$\log p(y^i = c \mid x^i) - \log p(y^i = c' \mid x^i)$



https://web.stanford.edu/~hastie/Papers/ESLII.pdf

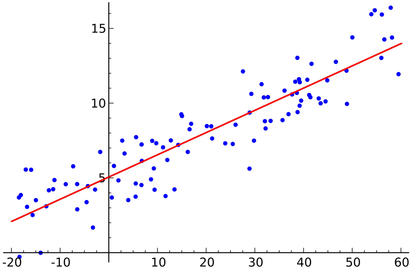
- This leads to a quadratic classifier.
  - GDA is sometimes called quadratic discriminant analysis.

Outline

1. Conditional Independence
2. Learning in Multivariate Gaussians
3. Supervised Learning with Gaussians
4. Bayesian Linear Regression
5. Rejection and Importance Sampling

Regression with Gaussians

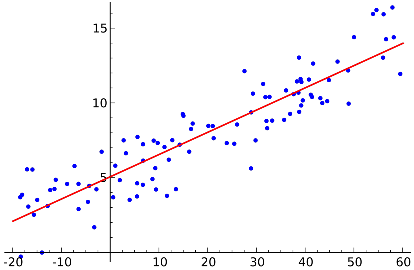
- Regression is a variant on supervised learning where y^i is continuous.



https://en.wikipedia.org/wiki/Regression_analysis

Regression with Gaussians

- Regression is a variant on supervised learning where y^i is continuous.

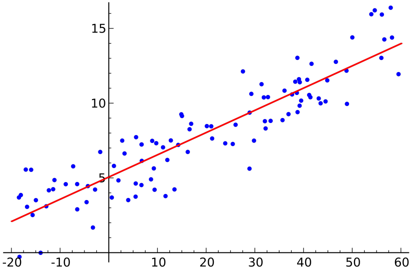


https://en.wikipedia.org/wiki/Regression_analysis

- It's possible to use generative regression models.
  - For example, we could model $p(x, y)$ as a multivariate Gaussian.
    - Then use that the conditional $p(y \mid x)$ is Gaussian for prediction.

Regression with Gaussians

- Regression is a variant on supervised learning where y^i is continuous.



https://en.wikipedia.org/wiki/Regression_analysis

- It's possible to use generative regression models.
  - For example, we could model $p(x, y)$ as a multivariate Gaussian.
    - Then use that the conditional $p(y \mid x)$ is Gaussian for prediction.

- But we usually treat features as fixed (as in discriminative classification models).
  - And to start, we will consider models that make linear predictions, $\hat{y}^i = w^\mathsf{T} x^i$.

L2-Regularized Least Squares and Gaussians

review

- A common linear regression model is L2-regularized least squares,

$$\mathop{\arg\min}_{w} \frac{1}{2\sigma^2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2.$$

L2-Regularized Least Squares and Gaussians

review

- A common linear regression model is L2-regularized least squares,

$$\underset{w}{\arg\min} \frac{1}{2\sigma^2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2.$$

- This corresponds to MAP estimation with a Gaussian likelihood and prior,

$$Y \sim \mathcal{N}(w^\top X, \sigma^2), \quad w \sim \mathcal{N}(0, \lambda^{-1}\mathbf{I}).$$

L2-Regularized Least Squares and Gaussians

review

- A common linear regression model is L2-regularized least squares,

$$\underset{w}{\arg\min}\, \frac{1}{2\sigma^2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2.$$

- This corresponds to MAP estimation with a Gaussian likelihood and prior,

$$Y \sim \mathcal{N}(w^\mathsf{T} X, \sigma^2), \quad w \sim \mathcal{N}(0, \lambda^{-1}\mathbf{I}).$$

- By setting the gradient to zero, the unique solution is given by:

$$\hat{w} = \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^\mathsf{T} y.$$

L2-Regularized Least Squares and Gaussians

review

- A common linear regression model is L2-regularized least squares,

$$\operatorname*{arg\,min}_{w} \frac{1}{2\sigma^2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2.$$

- This corresponds to MAP estimation with a Gaussian likelihood and prior,

$$Y \sim \mathcal{N}(w^\mathsf{T} X, \sigma^2), \quad w \sim \mathcal{N}(0, \lambda^{-1}\mathbf{I}).$$

- By setting the gradient to zero, the unique solution is given by:

$$\hat{w} = \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^\mathsf{T} y.$$

- In 340 we fixed $\sigma^2 = 1$ (since changing σ^2 is equivalent to changing λ).
  - In Bayesian inference, both σ^2 and λ affect the predictions.

L2-Regularized Least Squares and Gaussians

review

- A common linear regression model is L2-regularized least squares,

$$\underset{w}{\arg\min}\, \frac{1}{2\sigma^2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2.$$

- This corresponds to MAP estimation with a Gaussian likelihood and prior,

$$Y \sim \mathcal{N}(w^\mathsf{T} X, \sigma^2), \quad w \sim \mathcal{N}(0, \lambda^{-1}\mathbf{I}).$$

- By setting the gradient to zero, the unique solution is given by:

$$\hat{w} = \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^\mathsf{T} y.$$

- In 340 we fixed $\sigma^2 = 1$ (since changing σ^2 is equivalent to changing λ).
  - In Bayesian inference, both σ^2 and λ affect the predictions.
- To predict on new example $\tilde{x}$ with MAP estimate, we use $\hat{y} = \hat{w}^T\tilde{x}$.

Summary

- MLE for multivariate Gaussian:
  - MLE for μ is mean of data, MLE for Σ is covariance of data (if positive definite).
- Posterior and posterior predictive under Gaussian prior on mean is Gaussian.
  - Can be shown using that product of Gaussians is Gaussian.
- Graphical Lasso uses L1-regularization of precision matrix.
  - Leads to a sparse graph structure representing conditional independences.
- Supervised learning with Gaussians
  - Generative classifier with Gaussian classes is Gaussian discriminant analysis (GDA).
  - L2-regularized least squares is obtained using a Gaussian likelihood and prior.
    - Regression model assuming features fixed/non-random as in discriminative classifiers.

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- To get MLE for Σ we re-parameterize in terms of precision matrix $\Theta = \Sigma^{-1}$,

$$\frac{1}{2}\sum_{i=1}^{n}(x^i - \mu)^\top \Sigma^{-1}(x^i - \mu) + \frac{n}{2}\log|\Sigma|$$

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- To get MLE for Σ we re-parameterize in terms of precision matrix $\Theta = \Sigma^{-1}$,

$$
\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Sigma^{-1}(x^i-\mu)+\frac{n}{2}\log|\Sigma| \\
=&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Theta(x^i-\mu)+\frac{n}{2}\log|\Theta^{-1}| \qquad \text{(ok because } \Sigma \text{ is invertible)}
\end{aligned}
$$

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- To get MLE for Σ we re-parameterize in terms of precision matrix $\Theta = \Sigma^{-1}$,

$$
\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Sigma^{-1}(x^i-\mu)+\frac{n}{2}\log|\Sigma| \\
=&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Theta(x^i-\mu)+\frac{n}{2}\log|\Theta^{-1}| && \text{(ok because } \Sigma \text{ is invertible)} \\
=&\frac{1}{2}\sum_{i=1}^{n}\mathrm{Tr}\left((x^i-\mu)^\top\Theta(x^i-\mu)\right)+\frac{n}{2}\log|\Theta|^{-1} && (\text{scalar } y^\top Ay = \mathrm{Tr}(y^\top Ay))
\end{aligned}
$$

- Where the trace $\mathrm{Tr}(A)$ is the sum of the diagonal elements of A.

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- To get MLE for Σ we re-parameterize in terms of precision matrix $\Theta = \Sigma^{-1}$,

$$
\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Sigma^{-1}(x^i-\mu)+\frac{n}{2}\log|\Sigma| \\
=&\frac{1}{2}\sum_{i=1}^{n}(x^i-\mu)^\top\Theta(x^i-\mu)+\frac{n}{2}\log|\Theta^{-1}| && \text{(ok because } \Sigma \text{ is invertible)} \\
=&\frac{1}{2}\sum_{i=1}^{n}\text{Tr}\left((x^i-\mu)^\top\Theta(x^i-\mu)\right)+\frac{n}{2}\log|\Theta|^{-1} && \text{(scalar } y^\top Ay = \text{Tr}(y^\top Ay)\text{)} \\
=&\frac{1}{2}\sum_{i=1}^{n}\text{Tr}((x^i-\mu)(x^i-\mu)^\top\Theta)-\frac{n}{2}\log|\Theta| && (\text{Tr}(ABC)=\text{Tr}(CAB))
\end{aligned}
$$

- Where the trace $\text{Tr}(A)$ is the sum of the diagonal elements of A.
  - That $\text{Tr}(ABC)=\text{Tr}(CAB)$ when dimensions match is the cyclic property of trace.

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- From the last slide we have in terms of precision matrix Θ that

$$=\frac{1}{2}\sum_{i=1}^{n}\mathsf{Tr}((x^i-\mu)(x^i-\mu)^\top\Theta)-\frac{n}{2}\log|\Theta|$$

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- From the last slide we have in terms of precision matrix Θ that

$$=\frac{1}{2}\sum_{i=1}^{n}\mathsf{Tr}((x^i-\mu)(x^i-\mu)^\top\Theta)-\frac{n}{2}\log|\Theta|$$

- We can exchange the sum and trace (trace is a linear operator) to get,

$$=\frac{1}{2}\mathsf{Tr}\left(\sum_{i=1}^{n}(x^i-\mu)(x^i-\mu)^\top\Theta\right)-\frac{n}{2}\log|\Theta| \qquad \sum_i \mathsf{Tr}(A_iB)=\mathsf{Tr}\left(\sum_i A_iB\right)$$

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- From the last slide we have in terms of precision matrix Θ that

$$=\frac{1}{2}\sum_{i=1}^{n}\mathsf{Tr}((x^i-\mu)(x^i-\mu)^\top\Theta)-\frac{n}{2}\log|\Theta|$$

- We can exchange the sum and trace (trace is a linear operator) to get,

$$=\frac{1}{2}\mathsf{Tr}\left(\sum_{i=1}^{n}(x^i-\mu)(x^i-\mu)^\top\Theta\right)-\frac{n}{2}\log|\Theta| \qquad \sum_i \mathsf{Tr}(A_iB)=\mathsf{Tr}\left(\sum_i A_iB\right)$$

$$=\frac{n}{2}\mathsf{Tr}\left(\left(\underbrace{\frac{1}{n}\sum_{i=1}^{n}(x^i-\mu)(x^i-\mu)^\top}_{\text{sample covariance } `S'}\right)\Theta\right)-\frac{n}{2}\log|\Theta|. \qquad \left(\sum_i A_iB\right)=\left(\sum_i A_i\right)B$$

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- So the NLL in terms of the precision matrix Θ and sample covariance S is

$$f(\Theta) = \frac{n}{2}\text{Tr}(S\Theta) - \frac{n}{2}\log|\Theta|, \text{ with } S = \frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^\top$$

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- So the NLL in terms of the precision matrix Θ and sample covariance S is

$$f(\Theta) = \frac{n}{2}\text{Tr}(S\Theta) - \frac{n}{2}\log|\Theta|, \text{ with } S = \frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^\top$$

- Weird-looking but has nice properties:
  - $\text{Tr}(S\Theta)$ is linear function of Θ, with $\nabla_\Theta \text{Tr}(S\Theta) = S$.

(it's the matrix version of an inner-product $s^\top \theta$)

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- So the NLL in terms of the precision matrix Θ and sample covariance S is

$$f(\Theta) = \frac{n}{2}\text{Tr}(S\Theta) - \frac{n}{2}\log|\Theta|, \text{ with } S = \frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^\top$$

- Weird-looking but has nice properties:
  - $\text{Tr}(S\Theta)$ is linear function of Θ, with $\nabla_\Theta \text{Tr}(S\Theta) = S$.

    (it's the matrix version of an inner-product $s^\top\theta$)
  - Negative log-determinant is strictly convex, and has $\nabla_\Theta \log\det\Theta = \Theta^{-1}$.

    (generalizes $\nabla \log|x| = 1/x$ for for $x > 0$).

MLE for Multivariate Gaussians (Covariance Matrix)

bonus!

- So the NLL in terms of the precision matrix Θ and sample covariance S is

$$f(\Theta) = \frac{n}{2}\text{Tr}(S\Theta) - \frac{n}{2}\log|\Theta|, \text{ with } S = \frac{1}{n}\sum_{i=1}^{n}(x^i - \mu)(x^i - \mu)^\top$$

- Weird-looking but has nice properties:
  - $\text{Tr}(S\Theta)$ is linear function of Θ, with $\nabla_\Theta \text{Tr}(S\Theta) = S$.

    (it's the matrix version of an inner-product $s^\top\theta$)
  - Negative log-determinant is strictly convex, and has $\nabla_\Theta \log\det\Theta = \Theta^{-1}$.

    (generalizes $\nabla \log|x| = 1/x$ for for $x > 0$).
- Using these two properties the gradient matrix has a simple form:

$$\nabla f(\Theta) = \frac{n}{2}S - \frac{n}{2}\Theta^{-1}.$$

Trace Regularization and L1-regularization

bonus!

- A classic regularizer for Σ is to add a diagonal matrix to S and use

$$\Sigma = S + \lambda I,$$

which satisfies $\Sigma \succ 0$ because $S \succeq 0$ (eigenvalues at least λ).

Trace Regularization and L1-regularization

bonus!

- A classic regularizer for Σ is to add a diagonal matrix to S and use

$$\Sigma = S + \lambda I,$$

  which satisfies $\Sigma \succ 0$ because $S \succeq 0$ (eigenvalues at least λ).

- This corresponds to L1-regularization of diagonals of precision.

$$\begin{aligned} f(\Theta) &= \mathrm{Tr}(S\Theta) - \log|\Theta| + \lambda \sum_{j=1}^{d} |\Theta_{jj}| && \text{(Gauss. NLL plus L1 of diags)} \\ &= \mathrm{Tr}(S\Theta) - \log|\Theta| + \lambda \sum_{j=1}^{d} \Theta_{jj} && \text{(Diagonals of pos. def. matrix are } > 0\text{)} \end{aligned}$$

- Taking gradient and setting to zero gives $\Sigma = S + \lambda$.
  - But doesn't set to exactly zero as log-determinant term is too "steep" at 0.

Trace Regularization and L1-regularization

bonus!

- A classic regularizer for Σ is to add a diagonal matrix to S and use

$$\Sigma = S + \lambda I,$$

which satisfies $\Sigma \succ 0$ because $S \succeq 0$ (eigenvalues at least λ).

- This corresponds to L1-regularization of diagonals of precision.

$$\begin{aligned}
f(\Theta) &= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \sum_{j=1}^{d} |\Theta_{jj}| && \text{(Gauss. NLL plus L1 of diags)} \\
&= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \sum_{j=1}^{d} \Theta_{jj} && \text{(Diagonals of pos. def. matrix are } > 0\text{)} \\
&= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \text{Tr}(\Theta) && \text{(Definition of trace)}
\end{aligned}$$

- Taking gradient and setting to zero gives $\Sigma = S + \lambda$.
  - But doesn't set to exactly zero as log-determinant term is too "steep" at 0.

Trace Regularization and L1-regularization

bonus!

- A classic regularizer for Σ is to add a diagonal matrix to S and use

$$\Sigma = S + \lambda I,$$

which satisfies $\Sigma \succ 0$ because $S \succeq 0$ (eigenvalues at least λ).

- This corresponds to L1-regularization of diagonals of precision.

$$\begin{aligned} f(\Theta) &= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \sum_{j=1}^{d} |\Theta_{jj}| && \text{(Gauss. NLL plus L1 of diags)} \\ &= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \sum_{j=1}^{d} \Theta_{jj} && \text{(Diagonals of pos. def. matrix are } > 0\text{)} \\ &= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \text{Tr}(\Theta) && \text{(Definition of trace)} \\ &= \text{Tr}(S\Theta + \lambda\Theta) - \log|\Theta| && \text{(Linearity of trace)} \end{aligned}$$

- Taking gradient and setting to zero gives $\Sigma = S + \lambda$.
  - But doesn't set to exactly zero as log-determinant term is too "steep" at 0.

Trace Regularization and L1-regularization

bonus!

- A classic regularizer for Σ is to add a diagonal matrix to S and use

$$\Sigma = S + \lambda I,$$

which satisfies $\Sigma \succ 0$ because $S \succeq 0$ (eigenvalues at least λ).

- This corresponds to L1-regularization of diagonals of precision.

$$\begin{aligned}
f(\Theta) &= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \sum_{j=1}^{d} |\Theta_{jj}| && \text{(Gauss. NLL plus L1 of diags)}\\
&= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \sum_{j=1}^{d} \Theta_{jj} && \text{(Diagonals of pos. def. matrix are } > 0\text{)}\\
&= \text{Tr}(S\Theta) - \log|\Theta| + \lambda \text{Tr}(\Theta) && \text{(Definition of trace)}\\
&= \text{Tr}(S\Theta + \lambda\Theta) - \log|\Theta| && \text{(Linearity of trace)}\\
&= \text{Tr}((S + \lambda I)\Theta) - \log|\Theta| && \text{(Distributive law)}
\end{aligned}$$

- Taking gradient and setting to zero gives $\Sigma = S + \lambda$.
  - But doesn't set to exactly zero as log-determinant term is too "steep" at 0.